IN THE CLAIMS:

The current claims follow. For claims not marked as amended in this response, any difference in the claims below and the previous state of the claims is unintentional and in the nature of a typographical error.

1.   (Original)   A method for testing code, comprising:

receiving a test request;

sending executable program code, corresponding to the test request, to a client

system;

receiving a response from the client system indicating that the client system will

perform a test, and indicating that the client system was not being actively

used when the executable program code was sent.

2.   (Original)   The method of claim 1, wherein executable program code,

corresponding to the test request, is sent to multiple client systems.

3.   (Original)   The method of claim 1, further comprising retrieving a list of client

system identifiers, the client system identifiers indicating client systems to which

executable program code can be sent for testing.

4.   (Original)   A method for testing code in a client data processing system,

comprising:

receiving executable code from a server system in a client data processing system;

if the client data processing system is in an idle state when the executable code is

    received, then

    sending a response to the server system,

    testing at least a portion of the executable code, and

    sending test results to the server system.

5.   (Original)   The method of claim 4, wherein if the client data processing

system is not in an idle state when the executable code is received, then no

response is sent to the server and no testing is performed.

6.   (Original)   The method of claim 4, wherein the testing is a coverage analysis

test.

7.   (Original)   The method of claim 4, wherein the client data processing system

is in an idle state when no user is actively operating the client data processing

system.

8.   (Currently Amended) A data processing system ~~having at least~~ a processor and

accessible memory, ~~comprising:~~ <u>the data processing system configured to</u>

~~means for receiving~~ <u>receive</u> a test request;

~~means for sending~~ <u>send</u> executable program code, corresponding to the test

request, to a client system; <u>and</u>

~~means for receiving~~ <u>receive</u> a response from the client system indicating that the

client system will perform a test, and indicating that the client system was

not being actively used when the executable program code was sent.

9.   (Original)   The data processing system of claim 8, wherein executable

program code, corresponding to the test request, is sent to multiple client systems.

10.   (Currently Amended) The data processing system of claim 8, ~~further comprising~~

~~means for retrieving~~ <u>wherein the data processing system is further configured to</u>

<u>retrieve</u> a list of client system identifiers, the client system identifiers indicating

client systems to which executable program code can be sent for testing.

11. (Currently Amended) A data processing system ~~having at least~~ a processor and

accessible memory, ~~comprising:~~ <u>the data processing system configured to</u>

~~means for receiving~~ executable code from a server system in a client data

processing system; <u>and,</u>

~~means for,~~ if the client data processing system is in an idle state when the

executable code is received,

~~sending~~ <u>send</u> a response to the server system,

~~testing~~ <u>test</u> at least a portion of the executable code, and

~~sending~~ <u>send</u> test results to the server system.

12. (Original) The data processing system of claim 11, wherein if the client data

processing system is not in an idle state when the executable code is received,

then no response is sent to the server and no testing is performed.

13. (Original) The data processing system of claim 11, wherein the testing is a

coverage analysis test.

14. (Original) The data processing system of claim 11, wherein the client data

processing system is in an idle state when no user is actively operating the client

data processing system.

15. (Original) A computer program product tangibly embodied in a machine-readable medium, comprising:

instructions for receiving a test request;

instructions for sending executable program code, corresponding to the test request, to a client system;

instructions for receiving a response from the client system indicating that the client system will perform a test, and indicating that the client system was not being actively used when the executable program code was sent.

16. (Original) The computer program product of claim 15, wherein executable program code, corresponding to the test request, is sent to multiple client systems.

17. (Original) The computer program product of claim 15, further comprising instructions for retrieving a list of client system identifiers, the client system identifiers indicating client systems to which executable program code can be sent for testing.

18. (Original)   A computer program product tangibly embodied in a machine-readable

medium, comprising:

    instructions for receiving executable code from a server system in a client data

    processing system;

    instructions for, if the client data processing system is in an idle state when the

    executable code is received,

        sending a response to the server system,

        testing at least a portion of the executable code, and

        sending test results to the server system.

19. (Original)   The computer program product of claim 18, wherein if the client data

processing system is not in an idle state when the executable code is received, then no

response is sent to the server and no testing is performed.

20. (Original)   The computer program product of claim 18, wherein the testing is a

coverage analysis test.

21. (Original)   The computer program product of claim 18, wherein the client data

processing system is in an idle state when no user is actively operating the client data

processing system.